REMARKS / ARGUMENTS

Claims 1-11 remain pending in this application. No claims have been canceled or added.

Priority

Applicants appreciate the Examiner's acknowledgment of the claim for priority and safe receipt of the priority document.

35 U.S.C. §112

The claims have been amended to overcome the outstanding rejections under this section. The Examiner is hereby invited to contact the undersigned with any questions.

35 U.S.C. §103

Claims 1-3, 5-6 and 9-11 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Diefendorff ("Simultaneous Multithreading Exploits Instruction - and Thread-Level Parallelism"). Claims 4 and 7-8 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Diefendorff in view of Thayer (U.S. Patent No. 6,154,831). These rejections are traversed as follows.

The present invention is directed to a processor that maintains an execution order even if the execution order is specified by being allocated to different threads (i.e., having data dependency relationships among the threads). Fig. 13 shows one example of the present invention. When an instruction #10 MOV is allocated to thread 0 (TH0) and instruction #11 ADD is allocated to thread 1 (TH1), the execution order of thread 0 (TH0 becomes #00-10-20-30). In thread 2 (TH1) the execution order of #11 ADD is allocated despite the instruction of #10 MOV which should be executed first. A data dependency relationship cannot be maintained when instruction #11 ADD is executed prior to instruction #10 MOV.

The pending claims recite that the plurality of threads have changeable execution priorities, a thread synchronization number which indicates the progress level corresponding to the thread and the execution priority of the first thread is higher than that of the second thread when a synchronization number of the first thread is the same value as the synchronization number of the second thread. The Examiner's attention is directed to the specification on page 28, lines 14-24, for example. Here is it stated that the THj generating logic ensures that the data using thread may not pass the data defining thread. This is achieved by arranging that THj be equal to 0 when thread synchronization numbers IDj0 and IDk1 are identical. Thus, when the thread synchronization numbers are identical, the data defining thread is selected.

In other words, when the thread synchronization numbers are identical, the data defining thread including the instruction of #10 MOV (the first thread) is selected after making its priority higher. The data defining thread including an instruction of #11 ADD is not executed. Then, #11 ADD of the second thread does not pass instruction #10 MOV. As such, the execution order is maintained. In the claims, the thread synchronization number is defined as a number which indicates a progress level corresponding to one of the plurality of threads. This is based on the fact that the number of repeats RCj matching each instruction in a queue is assigned as a thread synchronization number IDjn (see specification, page 26, lines 4-7). The number of repeats of each instruction shows a progress level of the threads in the case that a plurality of execution of instructions are being repeated.

In order to maintain the above-mentioned data dependency relationship, the present invention prevents a second thread from progressing prior to a first thread by making the first thread's priority higher when the first and second threads have identical synchronization numbers. On the other hand, Diefendorff shows that regardless of the data dependency relationship, a higher priority is given to the execution proceeding thread (see page 4, lines 13-15). As such, the present invention has a different priority changing method than Diefendorff. Diefendorff does not disclose or suggest the changing of priority according to a degree of procession of the threads.

Conclusion

In view of the foregoing, Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

Respectfully submitted,

MATTINGLY, STANGER, MALUR & BRUNDIDGE, P.C.

By _____
Shrinath Malur
Reg. No. 34,663
(703) 684-1120